AIAA 2016-1059

A Space-Time Discontinuous-Galerkin Approach for Separated Flows

Scott M. Murman, Laslo T. Diosady[*], Anirban Garai[†], and Marco Ceze[‡]

NASA Ames Research Center, Moffett Field, CA, USA

Abstract

The motivation and goals for developing a space-time spectral-element Discontinuous-Galerkin solver for complex separated flows are discussed. The desire for spectral elements in space and time to leverage current and next-generation computing hardware and enable the development of novel subgrid-scale physical models for scale-resolving simulations at practical engineering Reynolds numbers is discussed. Timing results for hardware-optimized kernels are presented and demonstrate the ability of space-time spectral-elements to utilize a significant fraction of the available computing power of an Intel Xeon processor. A dynamic Variational Multiscale Method is developed and applied to the simulation of channel flow at $Re_\tau = 544$.

1 Introduction

With support from the NASA Revolutionary Computational Aerosciences (RCA) subproject, a research effort to investigate the use of adaptive high-order methods for scale-resolving simulations of separated flow has been undertaken (*cf.* [1–3]). This research code,

[*]Science and Technology Corporation

[†]Oak Ridge Associated Universities

[‡]Oak Ridge Associated Universities

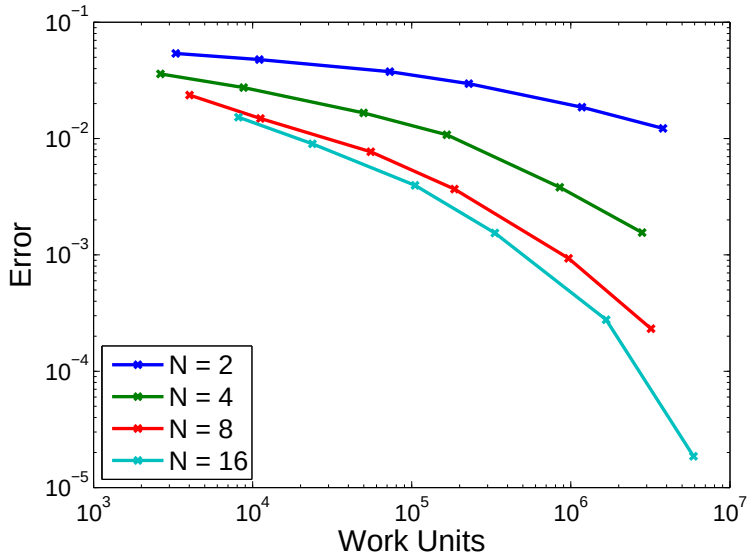


Figure 1: Truncation error in kinetic energy evolution for Taylor-Green vortex problem ($M_\infty = 0.1$, $Re = 1600$).

informally known as the eddy solver, is based on a space-time spectral-element Discontinuous-Galerkin (DG) algorithm. Recently we have begun collaborative partnerships to extend this effort to more direct engineering applications of interest in turbomachinery[4, 5], capsule wakes and parachute dynamics, and transonic aerodynamics for launch vehicles.

There are numerous similar research projects examining adaptive high-order DG methods for a variety of applications. The current work aims to provide some background for the motivation and goals of the current project, as well as technical details regarding the hardware optimization and physical modeling approach behind effort. Specifically, the term "high-order" is a catch-all for anything beyond a 2nd-order spatial operator. We specifically characterize our effort as a spectral-element approach to distinguish that we seek to approach spectral schemes in both space and time. Currently, we typically use 8th-order space-time elements in practical simulations. The use of spectral elements provides both opportunities for order-of-magnitude gains in computational efficiency over current state-of-the-art methods (*cf.* Fig. 1), and novel physical modeling approaches which take advantage of the scale separation available in a spectral representation. Both of these subjects are considered in terms of hardware-optimized algorithms and the development of a dynamic Variational Multiscale Method (VMM).

2 Background

A single optimal Computational Fluid Dynamics (CFD) tool for all situations does not exist, and seeking such an ultimate tool is a false goal. Rather, we have a Pareto frontier of optimality wherein tradeoffs are made based upon the particular needs and priorities of our targeted applications. For example, in solar physics simulations the emphasis is placed on capturing complex physical phenomena over large time integration windows in a spherical geometry, hence spectral methods are an attractive choice. A counterexample is aircraft conceptual design studies, where the emphasis on physics is suppressed and priority is given to automatic handling of arbitrary geometry, hence Cartesian methods are more appropriate. In order to discuss the technical approach behind the developement of the eddy solver, it is necessary to understand the context and goals behind the effort.

Within NASA our research supports a range of projects across the Aeronautics Research Mission Directorate (ARMD), through the Advanced Air Transport Technology (AATT) and Transformational Tools and Technologies (TTT) programs, the Human Exploration and Operations Mission Directorate (HEOMD), through the Space Launch System (SLS) and Orion programs, and the Space Technology Mission Directorate (STMD) through the Entry Systems Modeling (ESM) project. A sampling of applications of interest across these projects is presented in Fig. 2. With the exception of the parachute fluid-structure interaction, these simulations were produced using CFD tools developed within NASA, namely OVERFLOW, FUN3D, DPLR, and TURBO, using RANS modeling approaches. Our main goal is to supplement these existing tools, which perform well in many engineering settings but often struggle in the presence of complex unsteady physics.

The characteristics of the applications in Fig. 2 include complex geometry, complex physics (separation, shock-boundary layer interaction, jet interactions, aeroheating), multi-disciplinary analysis (parachute fluid-structure interaction), and multiphysics (combustion and non-equilibrium chemistry). These characteristics drive decisions about the type of numerical methods best suited to all of these diverse applications. The need to handle complex geometry leads to a desire for methods which accept adaptive unstructured meshes. The stiffness associated with high-Reynolds number flows, combustion, and chemistry motivates the use of fully implicit methods, so the timestep can be chosen by the physics encountered rather than stability limitations. The presence of massive separation, shock-boundary layer interaction, and jet interactions leads to a desire for scale-resolving modeling approaches such as large-eddy simulation (LES) to provide the necessary accuracy for the complex physics encountered. The computational expense associated with unsteady LES simulations

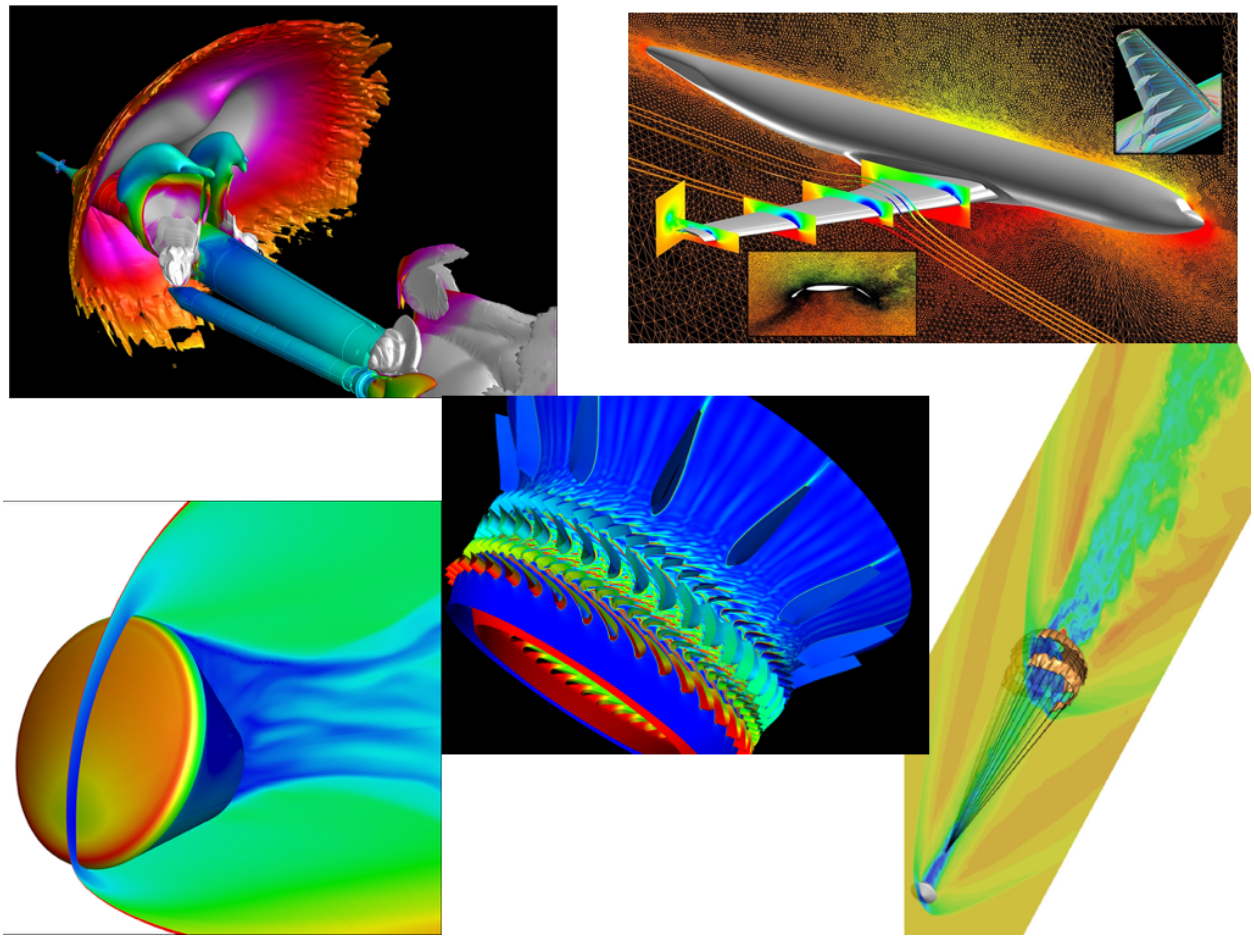


Figure 2: Sample of NASA CFD applications in aerodynamics, aeroheating, turbomachinery, and parachute fluid-structure interaction. Results computed using the OVERFLOW, FUN3D, DPLR, TURBO, and VTF solvers.

of high-Reynolds number applications motivates the extensive use of high-order methods in space and time in order to provide the necessary efficiency. The need for multiphysics and multi-disciplinary analysis requires a robust and flexible formulation that is easily extensible.

Examining the above requirements, we have chosen to develop a spectral-element DG formulation. This provides an efficient scheme on unstructured grids, which can utilize the VMM in order to provide a scale-resolving modeling approach[6]. We employ a fully implicit space-time scheme which enables the use of a robust entropy-stable formulation[7, 8] that extends consistently to the incompressible limit[9], as well as h and p adaptation in both space and time. An output-based h-p adaptive strategy based on an adjoint space-time formulation is being devloped[10]. Lastly, the Galerkin formulation provides a consistent mathematical foundation which has been demonstrated for fluid-structure interaction, combustion, and non-equilibrium chemistry applications(*cf.* [11–14]).

While we have chosen a DG algorithm, all of the above requirements can also be met using a Continuous-Galerkin (CG) algorithm. Initially the goal was to maintain both a DG and CG solver within the same computational framework, however this proved to be beyond our human resource limitations. Some of the rationale behind downselecting to a DG algorithm are the ease of implementation in an adaptive framework due to lower communication between elements, the ability to (optionally) utilize hanging nodes in an adaptive framework, and our focus on high-speed applications where robustness is primary and DG methods have a stronger track record.

3 Hardware-Optimized Kernels

NASA's main high-end computing (HEC) resource is the Pleiades Supercomputer, built around the family of Intel Xeon processors. The majority of legacy CFD solvers utilize less than 5% of the capacity of these processors in terms of single-threaded performance. For many years this inefficiency was hidden by gains in clock speed, however now clock speed is decreasing due to requirements to limit power consumption, and the legacy implementations are not well placed to take advantage of the latest chip designs. The cause of this behavior can be analyzed using a simplified roofline model[15] (Fig. 3). The horizontal axis is the operational intensity of the algorithm, essentially the number of flops performed for every byte of data movement. Finite-difference stencils have a low operational intensity ($\mathcal{O}(1)$), fast-fourier transform algorithms have a medium intensity ($\mathcal{O}(\log(N))$), and dense linear algebra kernels have a high intensity ($\mathcal{O}(N)$). Most legacy CFD applications have an intensity

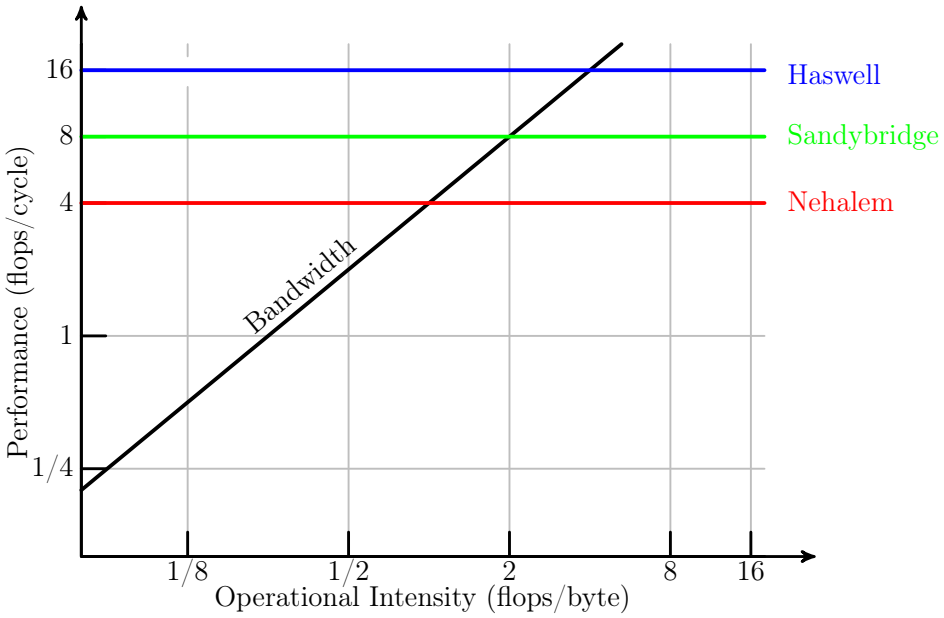


Figure 3: Simplified roofline model for Intel Xeon processors.

around 1/8, or one flop/double. This indicates that the performance of these algorithms is limited by memory bandwidth restrictions, irrespective of any optimization or tuning done to the code. As bandwidth restrictions are expected to be restrictive going forward, in order to take full advantage of current and future computational resources, algorithms with a high computational intensity are required.

While algorithms with high operational intensity may require more flops *per se*, the ability to leverage a large fraction of the processor capability more than counterbalances this cost. More importantly, the performance of high intensity operations will continue to improve in future processor generations, while lower intensity algorithms will suffer greater losses. Figure 4 presents an overview of the Intel Xeon processor roadmap. As noted above, clock speed is now essentially flat for each successive generation of processor. Floating-point performance gains are primarily being realized by wider single instruction, multiple data (SIMD) vector registers, and more complex instruction sets, *e.g.* fused multiply-add. It is often naively assumed that compiler optimization will automatically provide sufficient performance gains for new hardware, however Proebsting's Law[16] demonstrates that this is wildly optimistic, as compiler optimization doubles performance of a typical code roughly every 18 years. Given the complexity of evolving processor designs, this seems unlikely to improve going forward.

Space-time spectral-elements are well positioned to take advantage of modern processor

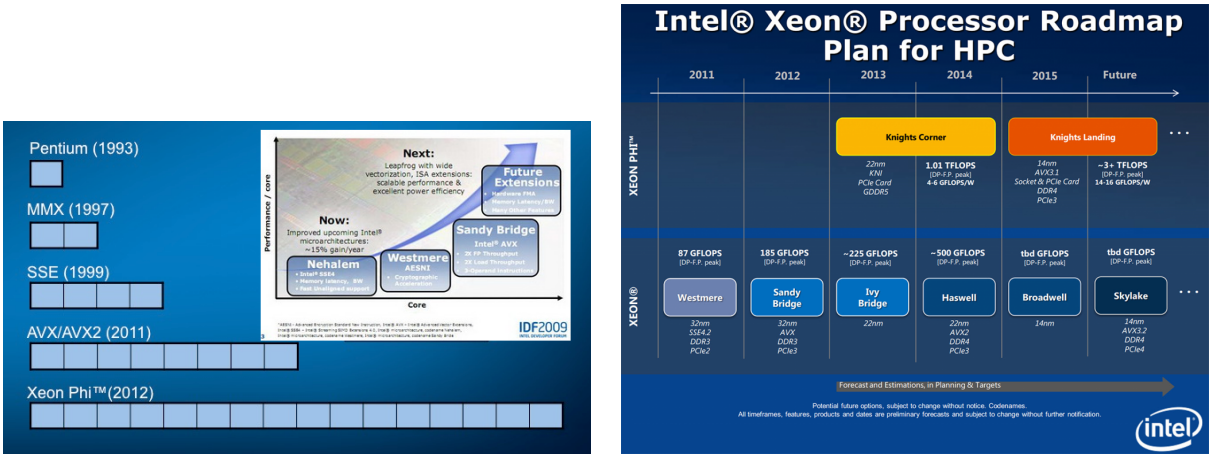


Figure 4: Overview of Intel Xeon processor roadmap.

designs, such as the Intel Xeon and Xeon Phi. A 16th-order element typically has over 1M coupled quadrature points,* so there is a lot of potential parallelism to exploit. We have designed our DG spectral-element solver around several core linear algebra kernels that are exposed in the tensor-product sum-factorization algorithm (*cf.* [1]). These kernels appear in the residual calculation, the matrix-free linearization, the preconditioner, *etc.* By optimizing these kernels specifically for the hardware of interest, in this case Xeon processors, the N^4 growth in the number of floating-point operations required with increasing order is offset. Figure 5 presents an example from [10], where the cost/dof for both the Jacobian linearization search direction and its transpose are essentially independent of solution order.†

Taking a deeper look, the performance of the linear algebra kernels are compared on the last two "ticks" of Intel Xeon processor models, Sandy Bridge and Haswell. Figure 6 presents benchmark timings for increasing order of accuracy using machine-coded routines and the Intel MKL BLAS library for comparison. It should be noted that the kernels are optimized for each architecture individually, *i.e.* the machine code for Sandy Bridge and Haswell are different, as is necessary to take advantage of the richer Intel AVX2 instruction set on Haswell. The benchmark in Fig. 6 represents roughly 20% of the computational cost of the eddy solver, and is hence a non-trivial demonstration. At 8th order the kernels achieve 75% and 70% of the processor theoretical peak on Sandy Bridge and Haswell respectively, and significantly outperform the MKL implementation. At 16th order cache effects appear and it is more difficult to outperform the BLAS library. The results in Fig. 6 demonstrate

*We typically evaluate quadratures using at least 2x the collocated dof.

†As this data is normalized by the spatial dof, an N^4 increase in cost would appear as linear growth.

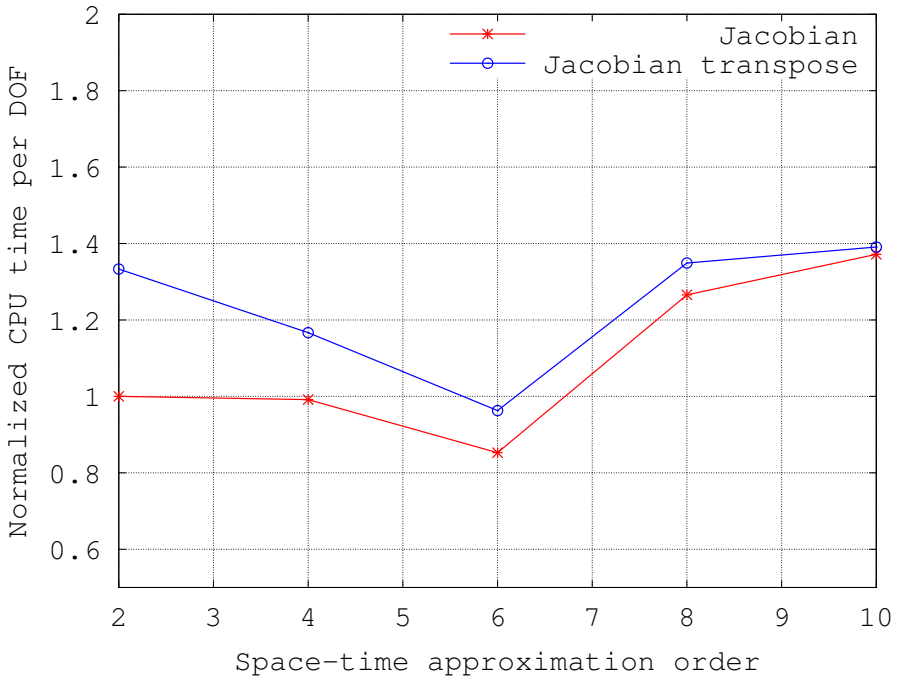
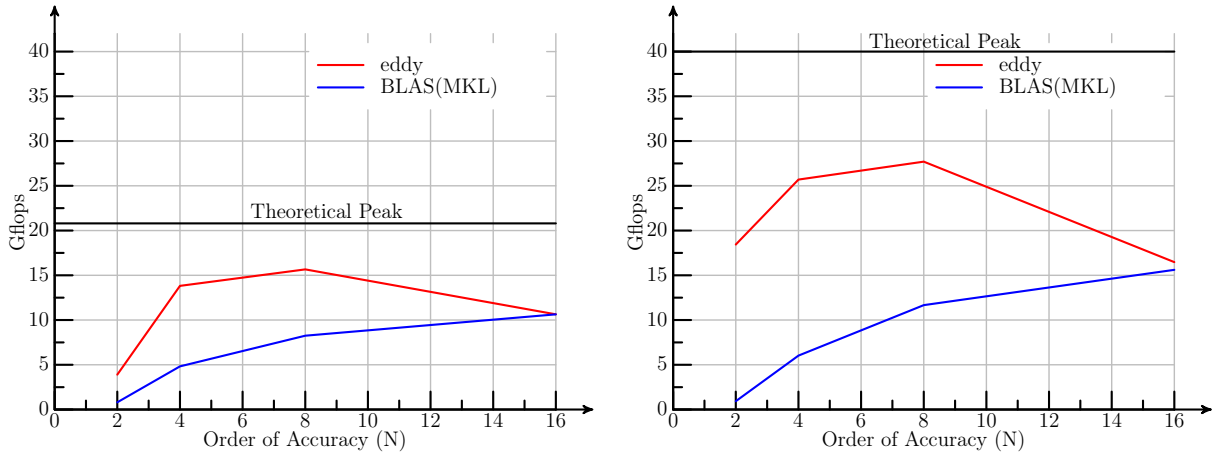


Figure 5: Computational cost per degree of freedom of the forward and transpose Fréchet derivatives – Computed on one processor core of Intel Xeon E5-2680v2.



(a) Sandy Bridge, 2.6 GHz **(b)** Haswell, 2.5 GHz

Figure 6: Performance of hardware-optimized tensor-product sum-factorization linear algebra kernels.

that the potential of spectral elements is not just theoretical. The increase in performance from Sandy Bridge to Haswell is roughly 2x across all polynomial degrees, in line with the increase in theoretical peak between the machines, even though the processor clock speed actually decreases. The above trends hold true on the next generation of Xeon processors, which provide another 2x increase in SIMD vector width.

The space-time formulation provides the opportunity to exploit multiple levels of parallelism. As is standard in most CFD codes, elements are distributed in space and communication between them uses the MPI protocol. Parallelization using SIMD instructions is carried out, as described above. In addition, the quadrature points in the temporal direction within each element can be processed in parallel, in this case using OpenMP threads. This time-parallel implementation is especially important for spectral-element methods, as the elements themselves are large when using very high order. To discretize using 1M space-time dof using 16th-order elements requires only 16 elements. It is necessary to scale beyond 16 MPI ranks in order to reduce the wallclock turnaround time.

The linear algebra kernel benchmark discussed above operates over spatial dof only. This is wrapped using an OpenMP threaded implementation for the temporal projection operator at each spatial dof and for each flow equation. The strong scaling performance on an Ivy Bridge node using this approach for 8th and 16th order elements is presented in Fig. 7. The benchmark achieves roughly 90% parallel efficiency on all cores within a node at 16th order. Combined with the results from Fig. 6, this provides a measured performance of roughly 500 Gflops per Haswell Xeon node for the full space-time benchmark at 8th order. For

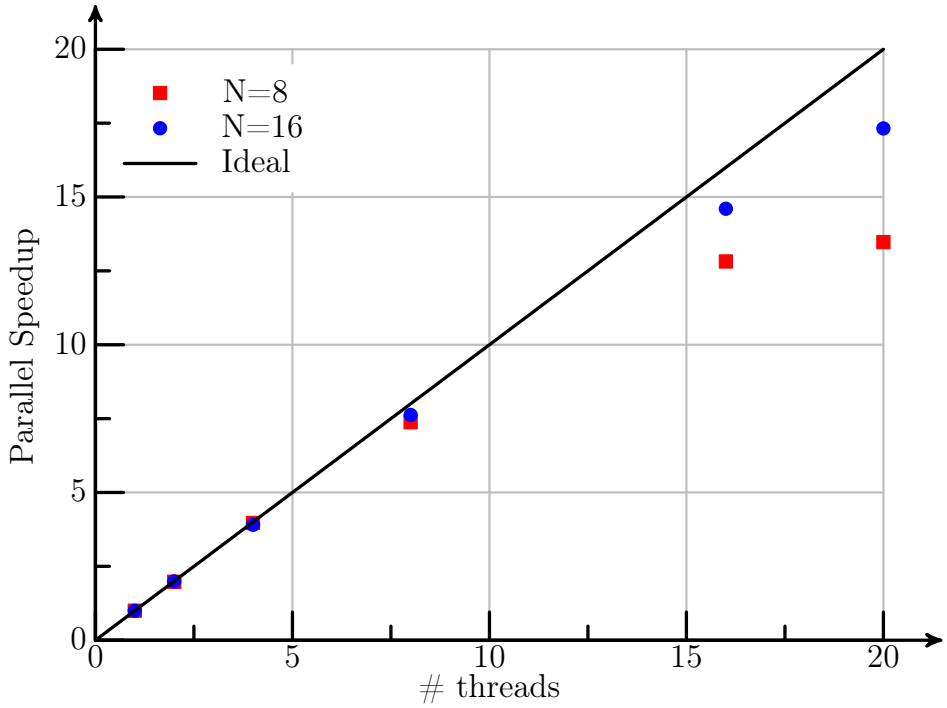


Figure 7: Computational cost per degree of freedom of the forward and transpose Fréchet derivatives – Computed on one processor core of Intel Xeon E5-2680v2.

comparison, the theoretical double-precision peak of an NVIDIA Tesla K40m GPU is 1.43 Tflops.

4 Dynamic Variational Multiscale Method

The VMM is a reformulation of the classical LES method, in which the filtering operation, used to explicitly separate resolved and unresolved scales, is replaced by a Galerkin projection operator [6, 17]. In this way the numerical method and the physical model are strongly integrated, *i.e.* the physical model is specifically tailored to take advantage of features of the numerical scheme. This philosophy is mirrored in the development of wavelet-based schemes[18] and compact Padé schemes[19] for scale-resolving methods.

In the VMM, the long-distance triadic interactions involving the unresolved scales are ignored, and the unresolved scales are assumed to only interact with the finest resolved scales, thus ensuring that no energy is removed from the large structures in the flow *via* a model. Previous work using VMM demonstrates the success of using a fixed-coefficient eddy-viscosity model for the sub-grid stresses, including for attached wall-bounded applications

[17, 20–22]. In order to apply the method to general complex flows, including those with separation, it is necessary to replace the fixed-coefficient model with a mechanism that automatically adapts to the local resolution and flow physics. As the typical VMM model is based on a Smagorinsky eddy-viscosity model, leveraging Germano's dynamic procedure from classical LES [23] is an obvious first step. This has been accomplished for finite-volume VMM schemes using an agglomeration operator to separate the fine and coarse scales[24].

Previous work developed a dynamic VMM approach for the current entropy-stable DG spectral-element solver[25]. The use of spectral elements provides an efficient scheme for resolving complex flows with a range of physical scales, and also a method that provides clear separation of these scales for the VMM. This dynamic VMM approach is first reviewed here for completeness, and then results of predicting a channel flow at $Re_\tau = 544$ are presented.

4.1 Incompressible Formulation

We begin by describing the dynamic approach for the incompressible Navier-Stokes equations to simplify the discussion, highlight the similarities to the classical LES approach, and demonstrate the parallels to the compressible formulation. Development of an entropy-stable compressible formulation then follows.

The incompressible, isothermal Navier-Stokes equations are given by

$$u_{i,i} = 0 \tag{1}$$

$$u_{i,t} + \left(u_i u_j\right)_{,j} = -\frac{1}{\rho}\partial_j\left(p\delta_{ij}\right) + \nu u_{i,jj}, \tag{2}$$

Writing the momentum equations in weak form over the domain Ω we have

$$\begin{aligned} &\left(u_{i,t}, w_i\right) - \left(u_i u_j, w_{i,j}\right) - \frac{1}{\rho}\left(p\delta_{ij}, w_{i,j}\right) + \left(\nu u_{i,j}, w_{i,j}\right) \\ &+ \left(u_i u_j + p\delta_{ij}, w_i n_j\right)_{\partial\Omega} - \left(\nu u_{i,j}, w_i n_j\right)_{\partial\Omega} = 0 \quad \forall w \in \mathcal{V}, \end{aligned} \tag{3}$$

where w is the test function. This is written compactly as

$$R_u\left(u, w\right) = 0. \tag{4}$$

where R_u is the bilinear operator of Eq. 3.

In a variational multiscale method we *a priori* decompose the continuous space as $\mathcal{V} = \tilde{\mathcal{V}} \cup \check{\mathcal{V}} \cup \hat{\mathcal{V}}$ where $\tilde{\mathcal{V}}$ are the coarse scales, $\check{\mathcal{V}}$ are the fine scales, and $\hat{\mathcal{V}}$ are the unresolved

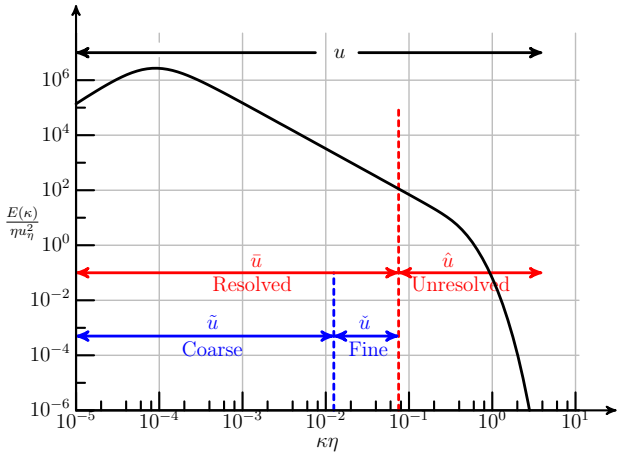


Figure 8: *A priori* scale separation for a variational multiscale method following the triple decomposition of [26].

scales that cannot be represented on the current discretization (*cf.* Figure 8, [26]). A similar decomposition follows for the velocity field, $u = \tilde{u}+\check{u}+\hat{u}$. Under suitable choice of orthogonal spaces, $\tilde{\mathcal{V}} \cap \check{\mathcal{V}} = \emptyset$, *etc.*, we have the following for the incompressible momentum equations

$$R_u\left(\bar{u}, \bar{w}\right) + \tau\left(u, \bar{w}\right) = 0 \quad \forall \bar{w} \in \bar{\mathcal{V}} = \tilde{\mathcal{V}} \cup \check{\mathcal{V}}, \tag{5}$$

where $\bar{u} = \tilde{u} + \check{u}$, $\bar{w} = \tilde{w} + \check{w}$, and

$$\tau\left(u, \bar{w}\right) = \left(\hat{u}_i\hat{u}_j, \bar{w}_{i,j}\right) + \left(\bar{u}_i\hat{u}_j, \bar{w}_{i,j}\right) + \left(\hat{u}_i\bar{u}_j, \bar{w}_{i,j}\right) \tag{6}$$

$$= \left(u_iu_j - \bar{u}_i\bar{u}_j, \bar{w}_{i,j}\right) \tag{7}$$

is the projection of the unresolved/"subgrid-scale" stress terms onto $\bar{w}$ that must be modeled to close the system.

The VMM assumes the unresolved scales only interact with the fine scales. The coarse and fine scales are defined by low-pass ($\mathbb{P}$) and high-pass ($\overline{\mathbb{P}}$) orthogonal projection operators respectively on the resolved scales,

$$\tilde{w} = \mathbb{P}\bar{w}, \quad \check{w} = \overline{\mathbb{P}}\bar{w}, \quad \overline{\mathbb{P}}\left(\mathbb{P}\bar{w}\right) = \emptyset, \quad \mathbb{P}\left(\overline{\mathbb{P}}\bar{w}\right) = \emptyset. \tag{8}$$

Assuming a gradient-diffusion (eddy-viscosity) model for τ we have

$$\tau\left(u, \bar{w}\right) \simeq -2\left(\left(C_1\Delta\right)^2 \|\check{S}_{i,j}\|\check{S}_{i,j}, \check{w}_{i,j}\right), \tag{9}$$

where $S_{i,j} = 1/2\left(u_{i,j} + u_{j,i}\right)$ is the symmetric strain-rate tensor. This is similar to the high-pass filtered Smagorinsky models[27]. The subgrid-stress coefficient C_1 is not equivalent to the standard Smagorinsky constant due to the scale separation of the VMM. The subgrid-stress coefficient is inside the inner product operator, which is akin to keeping the Smagorinsky constant inside the filter operator in a classical LES method.

Equation 9 is a bilinear operator valid over any space $\bar{\mathcal{V}}$, hence we can construct a variational analogue to the classical LES Germano dynamic procedure [23] to determine the eddy-viscosity coefficient. Denoting the current resolution by h, and applying a second projection (filter) operator to a coarser "test" space, H, we have the subgrid-stress on the current and test space as

$$\tau\left(u, \bar{w}^h\right) = \left(u_i u_j - \bar{u}_i^h \bar{u}_j^h, \bar{w}_{i,j}^h\right) \tag{10}$$

$$T\left(u, \bar{w}^H\right) = \left(u_i u_j - \bar{u}_i^H \bar{u}_j^H, \bar{w}_{i,j}^H\right). \tag{11}$$

Projecting the subgrid-stresses τ to the test space H and forming the variational dynamic Leonard stresses gives

$$L\left(u, \bar{w}^H\right) = T\left(u, \bar{w}^H\right) - \tau\left(u, \bar{w}^H\right) = \left(\bar{u}_i^h \bar{u}_j^h - \bar{u}_i^H \bar{u}_j^H, \bar{w}_{i,j}^H\right) = \\ -2\left(\left(C_1\Delta\right)^2 \|\check{S}_{i,j}^h\| \check{S}_{i,j}^h, \check{w}_{i,j}^H\right) + 2\left(\left(C_1\Delta\right)^2 \|\check{S}_{i,j}^H\| \check{S}_{i,j}^H, \check{w}_{i,j}^H\right), \tag{12}$$

where we've likewise followed a similar procedure for the modeled subgrid-stresses. Note that this approach varies from the dynamic localization of [28], whereby a variational formulation is built from the strong form of the Leonard stresses. Here a consistent formulation is used to directly construct a dynamic procedure for the variational form of the subgrid model stresses, similar to the approaches of [29] and [24].

If C_1 is assumed constant within an element, the inner product can be used to reduce Eq. 12 to a scalar equation with clear physical interpretation. For example, using the velocity as the test function, $\bar{w}_{i,j} = \bar{u}_{i,j}$, produces a variational analogue to Germano's dynamic procedure for classical LES. Similarly, defining

$$M_{ij} = -2\Delta^2 \|\check{S}_{i,j}^h\| \check{S}_{i,j}^h + 2\Delta^2 \|\check{S}_{i,j}^H\| \check{S}_{i,j}^H \tag{13}$$

and using the L_2 projection of M_{ij} for the test function reproduces a variational analogue of Lilly's least-square procedure [30].

4.2 Compressible Formulation

The Navier-Stokes equations for compressible viscous flow are

$$q_{i,t} + f_{ij,j} - g_{ij,j} = 0, \tag{14}$$

where q_i are the conserved quantities, f_{ij} is the inviscid flux, and g_{ij} is the viscous flux,

$$q_i = \begin{pmatrix} \rho \\ \rho u_k \\ \rho e \end{pmatrix}, \quad f_{ij} = \begin{pmatrix} \rho u_j \\ \rho u_k u_j + p\delta_{kj} \\ \rho e u_j + p u_j \end{pmatrix}, \quad g_{ij} = \begin{pmatrix} 0 \\ \tau_{kj} \\ u_k \tau_{kj} + \kappa T_{,j} \end{pmatrix}.$$

Writing Eq. 14 in weak form gives

$$(q_{i,t}, w_i) - (f_{ij}, w_{i,j}) + (g_{ij}, w_{i,j}) + (f_{ij}, w_i n_j)_{\partial\Omega} - (g_{i,j}, w_i n_j)_{\partial\Omega} = 0 \quad \forall w \in \mathcal{V} \tag{15}$$

or

$$R_q(q, w) = 0 \quad \forall w \in \mathcal{V}. \tag{16}$$

Again decomposing into fine, coarse, and unresolved scales, we assume that the effect of the unresolved scales on the diffusion coefficients is negligible. This gives

$$R_q(\bar{q}, \bar{w}) + \beta(q, \bar{w}) = 0 \quad \forall \bar{w} \in \bar{\mathcal{V}}, \tag{17}$$

where

$$\beta(q, \bar{w}) = (f_{ij}(q) - f_{ij}(\bar{q}), \bar{w}_{i,j}). \tag{18}$$

Introducing the generalized entropy variables that symmetrize the compressible Navier-Stokes equations (*cf.* [7], [8]),

$$v_i = \begin{pmatrix} -\frac{s}{\gamma-1} + \frac{\gamma+1}{\gamma-1} - \frac{\rho e}{p} \\ \frac{\rho u_k}{p} \\ -\frac{\rho}{p} \end{pmatrix}, \tag{19}$$

we then have

$$\beta(v, \bar{w}) \simeq (f_{ij}(v) - f_{ij}(\bar{v}), \bar{w}_{i,j}). \tag{20}$$

Numerically, entropy in compressible flow fills an analogous (though more complex) role to kinetic energy in incompressible flow. When simulating an incompressible flow we desire

a scheme with a global energy bound, whereas in compressible flow simulations we desire a scheme with a global entropy bound. When the velocity field is used as the test function in Eqn. 12 the momentum flux is an advection of kinetic energy. Analogously, choosing the entropy variables as the test function in Eq. 20 leads to advection of entropy for the resolved "stresses",

$$\left(f_{ij}, v_{i,j}\right) = \int_{\Omega} \partial_i \left(u_i \rho s\right) = \int_{\partial\Omega} \rho s u_i n_i \quad \text{compressible} \tag{21}$$

$$\left(u_i u_j, u_{j,i}\right) = \int_{\Omega} \partial_i \left(u_i \frac{u_j u_j}{2}\right) = \int_{\partial\Omega} \frac{u_j u_j}{2} u_i n_i \quad \text{incompressible} \tag{22}$$

Because the entropy variables are complex nonlinear functions of the conservative variables, the model integrated using entropy variables is not identical to the scale separation in conservative variables, *i.e.* $f_{ij}\left(\bar{v}\right) \neq f_{ij}\left(\bar{q}\right)$. A similar approximation under different modeling assumptions is described in [31]. The alternative, applying the scale separation directly to the entropy variables, introduces nonlinear products in the time derivative that are difficult to model. In the current low-speed numerical tests these approximations are unimportant. A more thorough analysis of the modeling assumptions for high-speed compressible flow is left for future work.

The subgrid-stress model for compressible flows mimics the diffusion terms in entropy variables,

$$\beta\left(v, \bar{w}\right) \simeq -\left(\left(C_1\Delta\right)^2 \|\check{v}_{i,j}\| \check{v}_{i,j}, \check{w}_{i,j}\right). \tag{23}$$

The dynamic procedure then becomes

$$\left(f_{ij}\left(\bar{v}^h\right) - f_{ij}\left(\bar{v}^H\right), \bar{w}^H_{i,j}\right) = -\left(\left(C_1\Delta\right)^2 \|\check{v}^h_{i,j}\| \check{v}^h_{i,j}, \check{w}^H_{i,j}\right) + \left(\left(C_1\Delta\right)^2 \|\check{v}^H_{i,j}\| \check{v}^H_{i,j}, \check{w}^H_{i,j}\right), \tag{24}$$

with the entropy variables used as the test function to determine the scalar coefficient C_1. Alternatively, by assuming a value for the subgrid-stress Prandtl number the modeled stresses can be constructed using the subgrid-stress analogue to the full viscous Jacobian instead of the diagonal approximation in Eq. 23. These two approaches are indistinguishable in the current low-speed testing.

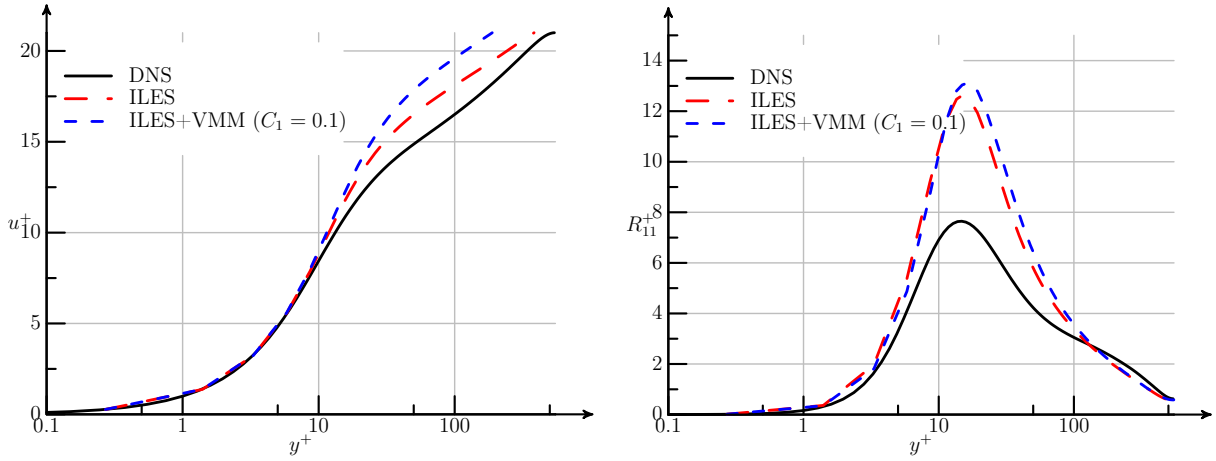
4.3 Channel $Re_\tau = 544$

A priori testing and analysis of the dynamic procedure for homogenous isotropic turbulence, channel flow, and flow over periodic hills is presented in [25]. Here we examine *a posteriori* performance for a channel flow at $Re_\tau = 544$. Simulations using 4th order in time and

8th order in space were performed using the same aspect ratio computational domain as Lee and Moser[32] ($8\pi h$ x $2h$ x $3\pi h$). Previous simulations demonstrate that the dynamic VMM converges to the DNS result for the channel flow given sufficient resolution, however this is simply a demonstration of consistency. Here, the near-wall spacing is intentionally coarser than typical wall-resolved LES simulations in order to highlight modeling differences. With sufficient resolution many subgrid-scale modeling approaches provide acceptable results, and this is straightforward to construct for model problems such as the current channel flow. In practical situations sufficient resolution is not available, and is not known *a priori*, hence we seek to test the behavior under realistic under-resolved conditions. The near-wall resolution of the current computational mesh is $\Delta t^+ = 1, \Delta x^+ = 100, \Delta y^+ = 1, \Delta z^+ = 50$, which is roughly an order of magnitude larger in the streamwise and span directions than the resolution used in Lee and Moser's spectral DNS computations. These quoted resolutions represent the average over the wall-bounded element.

The entropy-stable DG scheme is used to compute the statistics over 20 eddy turnover timescales using the Ismail and Roe approximate Riemann solver[33], which is effectively an implicit LES (ILES) model, and the same scheme using the VMM with a static coefficient, $C_1 = 0.1$. This coefficient matches the value used by Hughes *et al.* for $Re_\tau = 180$ channel flow simulations[20]. The high-pass VMM filter cuts off at the top 1/2 of the computed modes. Figure 9 presents the computed mean velocity and R_{11} Reynolds-stress profile after averaging over the homogeneous directions. The profiles indicate that both simulations are overly dissipative through the buffer and log layers, as is expected with insufficient resolution. The VMM simulations using a static coefficient negatively impact the results.

We use a model turbulent energy spectrum to help understand the behavior observed in Fig. 9. Idealized behavior using an ILES method and a dynamics LES method for two resolutions are presented in Fig. 10. In an ILES simulation the numerical dissipation is expected to act as a subgrid-scale model. Regardless of how well this numerical model behaves, by definition the ILES model is being applied on a mesh incapable of fully resolving the desired Reynolds number. Thus, the ILES is at best simulating a flow at an unknown lower Reynolds number depending upon the numerics of the method and the resolution. When we add a static-coefficient VMM we are exacerbating this situation and adding more dissipation that effectively lowers the resolved Reynolds number even further. In contrast, the goal of a dynamic LES procedure is to accurately resolve the turbulent energy spectrum through the inertial subrange scales, independent of mesh resolution. The modeled dissipation is then intended to mimic the physical processes associated with the smallest scales.



(a) Mean Velocity **(b)** R_{11} Reynolds stress

Figure 9: Computed velocity profiles for channel flow at $Re_\tau = 544$. DNS data from Lee and Moser[32].

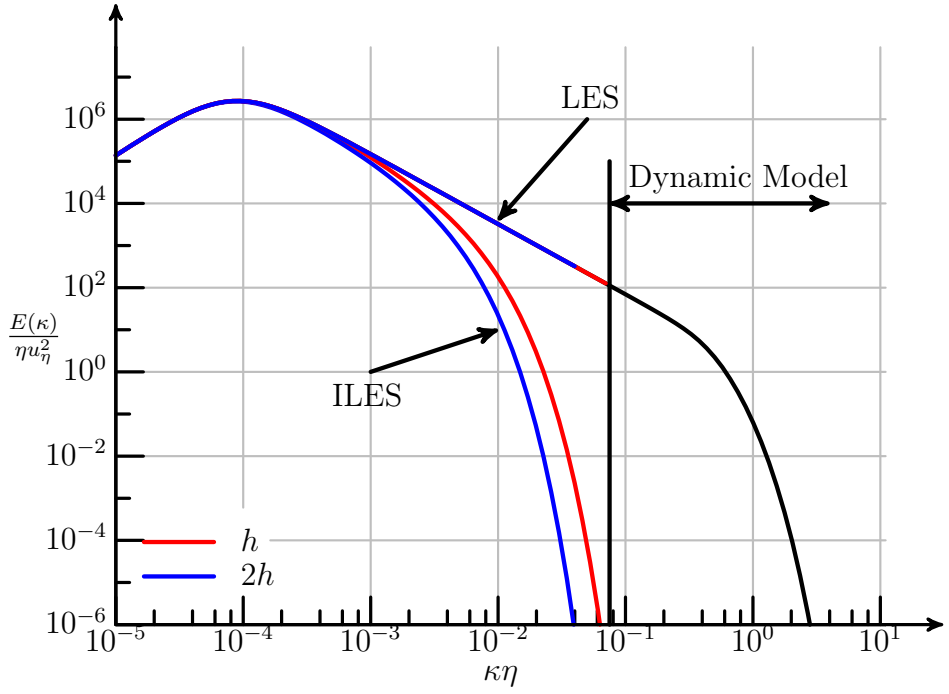


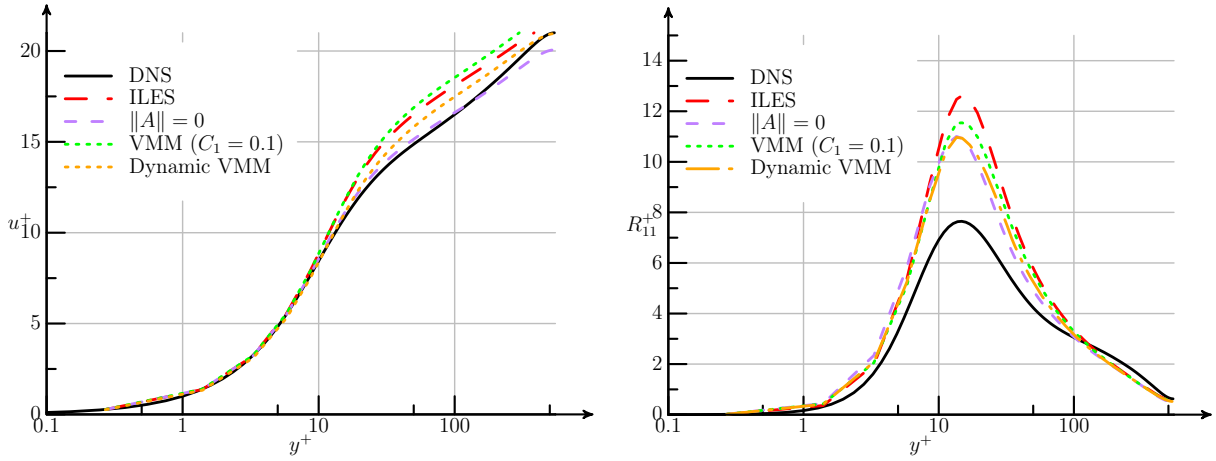
Figure 10: Idealized turbulent spectrum for ILES and dynamic LES simulations.

In order to properly capture the inertial subrange scales appropriate numerical methods must be utilized. For incompressible flows these methods are typically high-order skew symmetric operators, or similar, which are kinetic energy preserving and have no dissipation and low dispersion error (*cf.* [34, 35]). For compressible flows, the entropy-stable schemes are inherently dissipative, and hence an appropriate choice of method is unclear. Here, we've initially chosen to simply remove the numerical dissipation associated with the approximate Riemann solver. While this is not formally entropy stable independent of resolution, the dissipation associated with the physical subgrid-scale model is intended to provide sufficient stability. Note that the construction of the dynamic VMM described above is guaranteed not to remove entropy from the domain.

Repeating the channel flow simulations now with the numerical dissipation from the Riemann solver removed, and then using the same scheme with both the static VMM ($C_1 = 0.1$) and the dynamic VMM procedure is presented in Fig. 11. The "no dissipation" simulation is the best prediction of the DNS result, but still overpredicts the the production of Reynolds stress through the buffer layer at the current coarse resolution. The dynamic VMM procedure does add dissipation, however it is less than the ILES and static VMM dissipation, especially in terms of Reynolds stress production. In this sense the dynamic VMM procedure does a better job removing the sensitivity of subgrid-scale models to mesh resolution than the ILES or static coefficient VMM methods. This is important when applying these methods in a truly predictive sense for complex flows. The variation of mean subgrid-scale coefficient, C_1, across the channel half-width is presented in Fig. 12. As desired, the dynamic VMM coefficient approaches zero in the viscous near-wall region, and is in good agreement with previous analysis in the log layer as $C_1 \approx 0.1$ in this region. The slight decay towards the centerline is also observed in classical dynamic LES simulations of channel flow[36].

5 Future Work

While there is always room for improvement, the eddy solver is a fully capable prototype for space-time spectral-element DG applied to scale-resolving simulations of separated flows. The use of a tensor-product preconditioner with an entropy-stable space-time formulation provides an efficient and robust platform that can be extended to more complex physics and multi-disciplinary applications. Hardware-optimized kernels developed specifically for the tensor-product sum-factorization approach provide an existence proof that spectral-element methods are capable of exploiting modern processor designs. An interesting next step is



(a) Mean Velocity **(b)** R_{11} Reynolds stress

Figure 11: Computed velocity profiles for channel flow at $Re_\tau = 544$. Except for the ILES simulation, all computations have removed the numerical dissipation from the Riemann solver. DNS data from Lee and Moser[32].

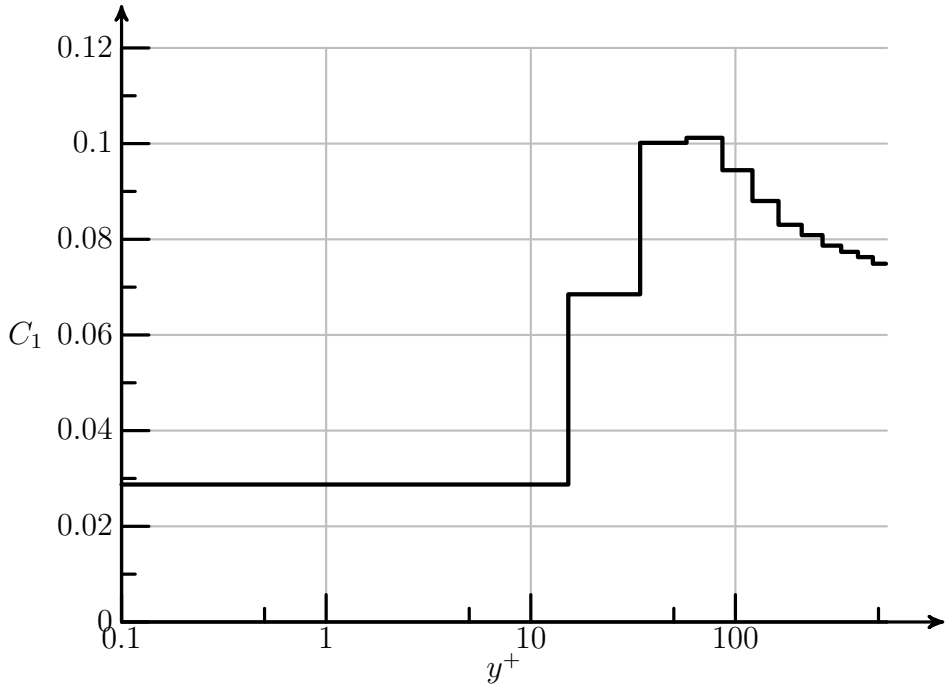


Figure 12: Variation of mean subgrid-scale coefficient with distance from wall over channel half-width. Coefficient is constant within an element.

examining how the time-parallel hybrid MPI/OpenMP approach extends to the novel Intel Xeon Phi Knights Landing architecture.

Currently we are partnering with subject-matter experts to apply these methods to relevant engineering applications of interest. The first demonstrations for low-pressure turbine blades are encouraging, and demonstrate the benefits of the approach over traditional 2nd-order RANS methods. Similar efforts are underway for capsule wake aerodynamics with parachute fluid-structure interaction, and transonic buffet of launch vehicles (*cf.* [37]). These efforts involve extending the space-time algorithm to a general relative motion scheme to support rotating machinery and parachute dynamics, as well as further developing the ability to handle high-order complex geometry (*cf.* [38]).

References

[1] L. Diosady and S. Murman, "Design of a Variational Multiscale Method for Turbulent Compressible Flows," AIAA Paper 2013-2870, June 2013.

[2] L. Diosady and S. Murman, "DNS of Flows over Periodic Hills using a Discontinuous Galerkin Spectral-Element Method," AIAA Paper 2014-2784, June 2014.

[3] Diosady, L.T. and Murman, S.M., "Higher-Order Methods for Compressible Turbulent Flows Using Entropy Variables," AIAA Paper 2105-0294, 2015.

[4] Garai, A., Diosady, L.T., Murman, S.M., and Madavan, N., "DNS of Flow in a Low-Pressure Turbine Cascade Using a Discontinuous-Galerkin Spectral-Element Method," in *Proceedings of ASME Turbo Expo 2015*, no. GT2015-42773, 2015.

[5] Garai, A., Diosady, L.T., Murman, S.M., and Madavan, N., "DNS of Flow in a Low-Pressure Turbine Cascade with Elevated Inflow Turbulence Using a Discontinuous-Galerkin Spectral-Element Method," in *Proceedings of ASME Turbo Expo 2016*, no. GT2016-56700, 2016.

[6] T. J. R. Hughes, G. R. Feijoo, L. Mazzei, and J.-B. Quincy, "The variational multiscale method - a paradigm for computational mechanics," *Comput. Methods Appl. Math.*, vol. 166, pp. 3–24, 1998.

[7] T. J. R. Hughes, L. Franca, and M. Mallet, "A new finite element formulation for computational fluid dynamics: I Symmetric forms of the compressible Euler and Navier-

Stokes equations and the second law of thermodynamics," *CMAME*, vol. 54, pp. 223–234, 1986.

[8] T. J. Barth, "Numerical methods for gasdynamic systems on unstructured meshes," in *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws* (D. Kroner, M. Olhberger, and C. Rohde, eds.), pp. 195–282, Springer Verlag, Berlin, 1999.

[9] Hauke, G. and Hughes, T.J.R., "A comparative study of different sets of variables for solving compressible and incompressible flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 153, pp. 1–44, 1998.

[10] Ceze, M., Diosady, L.T., and Murman, S.M., "Development of a High-Order Space-Time Matrix-Free Adjoint Solver," AIAA Paper 2016-0833, 2016.

[11] Oberlack, M. and Kummer, F., *Discontinuous Galerkin Methods for Premixed Combustion Multiphase Problems*, ch. Flow and Combustion in Advanced Gas Turbine Combustors, pp. 235–260. Springer Netherlands, 2012.

[12] Froehle, B. and Perrson, P.O., "A high-order discontinuous Galerkin method for fluid–structure interaction with efficient implicit–explicit time stepping," *Journal of Computational Physics*, vol. 272, no. 1, 2014.

[13] Lv, Y. and Imhe, M., "Discontinuous Galerkin method for multicomponent chemically reacting flows and combustion," *Journal of Computational Physics*, vol. 270, pp. 105–137, 2014.

[14] Michoski, C.E., Evans, J.A., and Schmitz, P.G., "Discontinuous Galerkin hp-adaptive methods for multiscale chemical reactors: Quiescent reactors," *Computer Methods in Applied Mechanics and Engineering*, pp. 163–197, 2014.

[15] Williams, S., Waterman, A., and Patterson, D., "Roofline: An Insightful Visual Performance Model for Floating-Point Programs and Multicore Architectures," *Communications of the ACM*, vol. 4, 2009.

[16] Scott, K., "On Proebsting's Law," University of Virginia Dept. of Computer Science Tech Report, 2012.

[17] T. J. Hughes, L. Mazzei, and K. E. Jansen, "Large eddy simulation and the variational multiscale method," *Computing and Visualization in Science*, vol. 3, pp. 47–59, 2000.

[18] Nejadmalayeri, A., Vezolainen, A., De Stafano, G., and Vasilyev, O.V., "Fully adaptive turbulence simulations based on Lagrangian spatio-temporally varying wavelet thresholding," *Journal of Fluid Mechanics*, vol. 749, pp. 794–817, 2014.

[19] Lele, S. K., "Compact Finite Difference Schemes with Spectral-like Resolution," *Journal of Computational Physics*, vol. 103, pp. 16–42, 1992.

[20] T. J. Hughes, A. A. Oberai, and L. Mazzei, "Large eddy simulation of turbulent channel flows by the variational multiscale method," *Physics of Fluids*, vol. 13, pp. 1784–1799, 2001.

[21] T. J. R. Hughes, G. N. Wells, and A. A. Wray, "Energy transfers and spectral eddy viscosity in large-eddy simulations of homogeneous isotropic turbulence: Comparison of dynamic smagorinsky and multiscale models over a range of discretizations," *Physics of Fluids*, vol. 16, no. 11, pp. 4044–4052, 2004.

[22] Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, and G. Scovazzi, "Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 14, pp. 173 – 201, 2007.

[23] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, "A dynamic subgrid-scale eddy viscosity model," *Physics of Fluids A*, vol. 3, no. 7, pp. 1760–1765, 1991.

[24] C. Farhat, J. Rajasekharan, and B. Koobus, "A dynamic variational multiscale method for large eddy simulations on unstructured meshes," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 13, pp. 1667 – 1691, 2006.

[25] S. Murman, L. Diosady, and A. Garai, "Development of dynamic sub-grid models for variational multiscale methods," in *Center for Turbulence Research Proceedings of the Summer Program 2014*, September 2014.

[26] S. S. Collis, "Monitoring unresolved scales in multiscale turbulence modeling," *Phys. Fluids*, vol. 13, pp. 1800–1806, 2001.

[27] S. Stolz, P. Schlatter, D. Meyer, and L. Kleiser, "High-Pass Filtered Eddy-Viscosity Models for LES," in *Direct and Large-Eddy Simulation V* (R. Friedrich, B. Geurts, and O. Métais, eds.), vol. 9 of *ERCOFTAC Series*, Springer Netherlands, 2004.

[28] Ghosal, S., Lund, T.S., Moin, P., and Akselvoll, K., "A dynamic localization model for large-eddy simulation of turbulent flows," *Journal of Fluid Mechanics*, vol. 286, pp. 229–255, 1995.

[29] Oberai, A. A. and Wanderer, J., "A dynamic multiscale viscosity method for the spectral approximation of conservation laws," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, pp. 1178–1792, 2006.

[30] D. Lilly, "A proposed modification of the germano subgrid-scale closure method," *Physics of Fluids A*, vol. 4, no. 3, pp. 633–635, 1991.

[31] Leveasseur, V., Sagaut, P., Chalot, F. and Davroux, A., "An entropy-variable-based VMS/GLS method for the simulation of compressible flows on unstructured grids," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, pp. 1154–1179, 2006.

[32] Lee, M., and Moser, R.D., "Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$," *Journal of Fluid Mechanics*, vol. 774, pp. 395–415, 2015.

[33] Ismail, F. and Roe, P.L., "Affordable, Entropy-consistent Euler flux functions II: entropy production at shocks," *Journal of Computational Physics*, vol. 228, no. 15, pp. 5410–5436, 2009.

[34] Morinishi, Y., Lund, T.S., Vasilyev, O.V., and Moin, P., "Fully conservative higher order finite differences for incompressible flows," *Journal of Computational Physics*, vol. 143, pp. 90–124, 1998.

[35] Kok, J.C., "A Symmetry and Dispersion-Relation Preserving High-Order Scheme for Aeroacoustics and Aerodynamics," ECCOMAS CFD, 2006.

[36] Park, N. and Mahesh, K., "Reduction of the Germano-identity error in the dynamic Smagorinsky model," *Physics of Fluids*, vol. 21, no. 6, 2009.

[37] Murman, S.M. and Diosady, L.T., "Simulation of a Hammerhead Payload Fairing in the Transonic Regime," AIAA Paper 2016-1548, 2016.

[38] Diosady, L.T. and Murman, S.M., "General element shapes within a tensor-product higher-order space-time discontinuous-Galerkin formulation," AIAA Paper 2105-3044, 2015.